

# Examples from Multilevel Software Comparative Reviews

Douglas Bates  
R Development Core Team  
Douglas.Bates@R-project.org

February 2005, with updates up to June 10, 2026

## **Abstract**

The Center for Multilevel Modelling at the Institute of Education, London maintains a web site of “Software reviews of multilevel modeling packages”. The data sets discussed in the reviews are available at this web site. We have incorporated these data sets in the `mlmRev` package for R and, in this vignette, provide the results of fitting several models to these data sets.

## **1 Introduction**

R is an Open Source implementation of John Chambers’ S language language for data analysis and graphics. R was initially developed by Ross Ihaka and Robert Gentleman of the University of Auckland and now is developed and maintained by an international group of statistical computing experts.

In addition to being Open Source software, which means that anyone can examine the source code to see exactly how the computations are being carried out, R is freely available from a network of archive sites on the Internet. There are precompiled versions for installation on the Windows operating system, Mac OS X and several distributions of the Linux operating system. Because the source code is available those interested in doing so can compile their own version if they wish.

R provides an environment for interactive computing with data and for graphical display of data. Users and developers can extend the capabilities of

R by writing their own functions in the language and by creating packages of functions and data sets. Many such packages are available on the archive network called CRAN (Comprehensive R Archive Network) for which the parent site is <https://cran.r-project.org>. One such package called `lme4` (along with a companion package called `Matrix`) provides functions to fit and display linear mixed models and generalized linear mixed models, which are the statisticians' names for the models called multilevel models or hierarchical linear models in other disciplines. The `lattice` package provides functions to generate several high level graphics plots that help with the visualization of the types of data to which such models are fit. Finally, the `mlmRev` package provides the data sets used in the "Software Reviews of Multilevel Modeling Packages" from the Multilevel Modeling Group at the Institute of Education in the UK. This package also contains several other data sets from the multilevel modeling literature.

The software reviews mentioned above were intended to provide comparison of the speed and accuracy of many different packages for fitting multilevel models. As such, there is a standard set of models that fit to each of the data sets in each of the packages that were capable of doing the fit. We will fit these models for comparative purposes but we will also do some graphical exploration of the data and, in some cases, discuss alternative models.

We follow the general outline of the previous reviews, beginning with simpler structures and moving to the more complex structures. Because the previous reviews were performed on older and considerably slower computers than the ones on which this vignette will be compiled, the timings produced by the `system.time` function and shown in the text should not be compared with previous timings given on the web site. They are an indication of the times required to fit such models to these data sets on recent computers with processors running at around 2 GHz or faster.

## 2 Two-level normal models

In the multilevel modeling literature a two-level model is one with two levels of random variation; the per-observation noise term and random effects which are grouped according to the levels of a factor. We call this factor a *grouping factor*. If the response is measured on a continuous scale (more or less) our initial models are based on a normal distribution for the per-observation noise and for the random effects. Thus such a model is called a "two-level normal

model” even though it has only one grouping factor for the random effects.

## 2.1 The Exam data

The data set called `Exam` provides the normalized exam scores attained by 4,059 students from 65 schools in inner London. Some of the covariates available with this exam score are the school the student attended, the sex of the student, the school gender (boys, girls, or mixed) and the student’s result on the Standardised London Reading test.

The R functions `str` and `summary` can be used to examine the structure of a data set (or, in general, any R object) and to provide a summary of an object.

```
> str(Exam)
'data.frame':      4059 obs. of  10 variables:
 $ school  : Factor w/ 65 levels "1","2","3","4",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ normexam: num  0.261 0.134 -1.724 0.968 0.544 ...
 $ schgend : Factor w/ 3 levels "mixed","boys",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ schavg  : num  0.166 0.166 0.166 0.166 0.166 ...
 $ vr      : Factor w/ 3 levels "bottom 25%","mid 50%",...: 2 2 2 2 2 2 2 2 2 2 ...
 $ intake  : Factor w/ 3 levels "bottom 25%","mid 50%",...: 1 2 3 2 2 1 3 2 2 3 ...
 $ standLRT: num  0.619 0.206 -1.365 0.206 0.371 ...
 $ sex     : Factor w/ 2 levels "F","M": 1 1 2 1 1 2 2 1 2 ...
 $ type    : Factor w/ 2 levels "Mxd","Sngl": 1 1 1 1 1 1 1 1 1 1 ...
 $ student : Factor w/ 650 levels "1","2","3","4",...: 143 145 142 141 138 155 158 115 117 113 ...

> summary(Exam)

  school      normexam      schgend      schavg
14      : 198   Min.      :-3.6660720  mixed:2169  Min.      :-0.75596
17      : 126   1st Qu.: -0.6995050  boys : 513   1st Qu.: -0.14934
18      : 120   Median   : 0.0043222  girls:1377  Median   :-0.02020
49      : 113   Mean     :-0.0001138                Mean     : 0.00181
8       : 102   3rd Qu.: 0.6787592                3rd Qu.: 0.21053
15      : 91    Max.     : 3.6660912                Max.     : 0.63766
(Other):3309

      vr      intake      standLRT      sex      type
bottom 25%: 640  bottom 25%:1176  Min.      :-2.93495  F:2436  Mxd :2169
mid 50%   :2263  mid 50%   :2344   1st Qu.: -0.62071  M:1623  Sngl:1890
top 25%   :1156  top 25%   : 539   Median   : 0.04050
                                      Mean     : 0.00181
                                      3rd Qu.: 0.61906
                                      Max.     : 3.01595

  student
20      : 34
54      : 34
15      : 33
22      : 33
31      : 33
59      : 33
(Other):3859
```

## 2.2 Model fits and timings

The first model to fit to the Exam data incorporates fixed-effects terms for the pretest score, the student's sex and the school gender. The only random-effects term is an additive shift associated with the school.

```
> (Em1 <- lmer(normexam ~ standLRT + sex + schgend + (1|school), Exam))
Linear mixed model fit by REML ['lmerMod']
Formula: normexam ~ standLRT + sex + schgend + (1 | school)
Data: Exam
REML criterion at convergence: 9347.674
Random effects:
Groups   Name             Std.Dev.
school   (Intercept) 0.293
Residual                    0.750
Number of obs: 4059, groups: school, 65
Fixed Effects:
(Intercept)      standLRT           sexM   schgendboys  schgendgirls
-0.001049      0.559754       -0.167392    0.177691    0.158997
```

The `system.time` function can be used to time the execution of an R expression. It returns a vector of five numbers giving the user time (time spent executing applications code), the system time (time spent executing system functions called by the applications code), the elapsed time, and the user and system time for any child processes. The first number is what is commonly viewed as the time required to do the model fit. (The elapsed time is unsuitable because it can be affected by other processes running on the computer.) These times are in seconds. On modern computers this fit takes only a fraction of a second.

```
> system.time(lmer(normexam ~ standLRT + sex + schgend + (1|school), Exam))
 user system elapsed
0.030  0.000  0.029
```

## 2.3 Interpreting the fit

As can be seen from the output, the default method of fitting a linear mixed model is restricted maximum likelihood (REML). The estimates of the variance components correspond to those reported by other packages as given on the Multilevel Modelling Group's web site. Note that the estimates of the variance components are given on the scale of the variance and on the scale of the standard deviation. That is, the values in the column headed `Std.Dev.` are simply the square roots of the corresponding entry in the `Variance` column. They are **not** standard errors of the estimate of the variance.

The estimates of the fixed-effects are different from those quoted on the web site because the terms for `sex` and `schgend` use a different parameterization than in the reviews. Here the reference level of `sex` is female (F) and the coefficient labelled `sexM` represents the difference for males compared to females. Similarly the reference level of `schgend` is `mixed` and the two coefficients represent the change from mixed to boys only and the change from mixed to girls only. The value of the coefficient labelled `Intercept` is affected by both these changes as is the value of the REML criterion.

To reproduce the results obtained from other packages, we must change the reference level for each of these factors.

```
> Exam$sex      <- relevel(Exam$sex, "M")
> Exam$schgend <- relevel(Exam$schgend, "girls")
> (Em2 <- lmer(normexam ~ standLRT + sex + schgend + (1|school), Exam))
Linear mixed model fit by REML ['lmerMod']
Formula: normexam ~ standLRT + sex + schgend + (1 | school)
Data: Exam
REML criterion at convergence: 9347.674
Random effects:
 Groups   Name      Std.Dev.
 school  (Intercept)  0.293
 Residual                    0.750
Number of obs: 4059, groups:  school, 65
Fixed Effects:
 (Intercept)      standLRT          sexF  schgendmixed  schgendboys
  -0.009444      0.559754      0.167392     -0.158997      0.018694
```

The coefficients now correspond to those in the tables on the web site. It happens that the REML criterion at the optimum in this fit is the same as in the previous fit, but you cannot depend on this occurring. In general the value of the REML criterion at the optimum depends on the parameterization used for the fixed effects.

## 2.4 Further exploration

### 2.4.1 Checking consistency of the data

It is important to check the consistency of data before trying to fit sophisticated models. One should plot the data in many different ways to see if it looks reasonable and also check relationships between variables.

For example, each observation in these data is associated with a particular student. The variable `student` is not a unique identifier of the student as it only has 650 unique values. It is intended to be a unique identifier within a school but it is not. To show this we create a factor that is the interaction of school and student then drop unused levels.

```

> Exam <- within(Exam, ids <- factor(school:student))
> str(Exam)
'data.frame':      4059 obs. of  11 variables:
 $ school  : Factor w/ 65 levels "1","2","3","4",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ normexam: num  0.261 0.134 -1.724 0.968 0.544 ...
 $ schgend  : Factor w/ 3 levels "girls","mixed",...: 2 2 2 2 2 2 2 2 2 2 ...
 $ schavg   : num  0.166 0.166 0.166 0.166 0.166 ...
 $ vr       : Factor w/ 3 levels "bottom 25%","mid 50%",...: 2 2 2 2 2 2 2 2 2 2 ...
 $ intake   : Factor w/ 3 levels "bottom 25%","mid 50%",...: 1 2 3 2 2 1 3 2 2 3 ...
 $ standLRT: num  0.619 0.206 -1.365 0.206 0.371 ...
 $ sex      : Factor w/ 2 levels "M","F": 2 2 1 2 2 1 1 1 2 1 ...
 $ type     : Factor w/ 2 levels "Mxd","Sngl": 1 1 1 1 1 1 1 1 1 1 ...
 $ student  : Factor w/ 650 levels "1","2","3","4",...: 143 145 142 141 138 155 158 115 117 113 ...
 $ ids      : Factor w/ 4055 levels "1:1","1:4","1:6",...: 48 49 47 46 45 50 51 39 40 38 ...

```

Notice that there are 4059 observations but only 4055 unique levels of student within school. We can check the ones that are duplicated

```

> as.character(Exam$ids[which(duplicated(Exam$ids))])
[1] "43:86" "50:39" "52:2"  "52:21"

```

One of these cases

```

> subset(Exam, ids == '43:86')
  school normexam schgend schavg vr intake standLRT sex type
2758   43 -0.8526700 mixed 0.4334322 top 25% mid 50% 0.1231502 M Mxd
2759   43  0.8219882 mixed 0.4334322 top 25% top 25% -0.0421520 F Mxd
  student ids
2758     86 43:86
2759     86 43:86

> xtabs(~ sex + school, Exam, subset = school %in% c(43, 50, 52), drop = TRUE)
  school
sex 43 50 52
 M  1 35 61
 F 60 38  0

```

is particularly interesting. Notice that one of the students numbered 86 in school 43 is the only male student out of 61 students from this school who took the exam. It is quite likely that this student's score was attributed to the wrong school and that the school is in fact a girls-only school, not a mixed-sex school.

The causes of the other three cases of duplicate student numbers within a school are not as clear. It would be necessary to go back to the original data records to check these.

The cross-tabulation of the students by sex and school for the mixed-sex schools

```

> xtabs(~ sex + school, Exam, subset = type == "Mxd", drop = TRUE)

```

|     |  | school |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |    |    |    |
|-----|--|--------|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|----|----|----|
| sex |  | 1      | 3  | 4  | 5  | 9  | 10 | 12 | 13 | 14  | 15 | 17 | 19 | 20 | 22 | 23 | 26 | 28 | 32 | 33 |
| M   |  | 45     | 29 | 45 | 16 | 21 | 31 | 23 | 26 | 92  | 47 | 31 | 33 | 21 | 48 | 10 | 44 | 31 | 27 | 44 |
| F   |  | 28     | 23 | 34 | 19 | 13 | 19 | 24 | 38 | 106 | 44 | 95 | 22 | 18 | 42 | 18 | 31 | 26 | 15 | 33 |

|     |  | school |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |
|-----|--|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--|--|
| sex |  | 34     | 38 | 42 | 43 | 45 | 46 | 47 | 50 | 51 | 54 | 55 | 56 | 59 | 61 | 62 | 63 |  |  |
| M   |  | 18     | 31 | 35 | 1  | 5  | 47 | 81 | 35 | 26 | 4  | 26 | 16 | 30 | 35 | 43 | 13 |  |  |
| F   |  | 8      | 23 | 23 | 60 | 48 | 36 | 1  | 38 | 32 | 4  | 25 | 22 | 17 | 29 | 28 | 17 |  |  |

shows another anomaly. School 47 is similar to school 43 in that, although it is classified as a mixed-sex school, 81 male students and only one female student took the exam. It is likely that the school was misrecorded for this one female student and the school is a male-only school.

Another school is worth noting. There were only eight students from school 54 who took the exam so any within-school estimates from this school will be unreliable.

A mosaic plot (Figure 1) produced with

```
> ExamMxd <- within(subset(Exam, type == "Mxd"), school <- factor(school))
> mosaicplot(~ school + sex, ExamMxd)
```

helps to detect mixed-sex schools with unusually large or unusually small ratios of females to males taking the exam.

## 2.4.2 Preliminary graphical displays

In addition to the pretest score (`standLRT`), the predictor variables used in this model are the student's sex and the school gender, which is coded as having three levels. There is some redundancy in these two variables in that all the students in a boys-only school must be male. For graphical exploration we convert from `schgend` to `type`, an indicator of whether the school is a mixed-sex school or a single-sex school, and plot the response versus the pretest score for each combination of sex and school type.

This plot is created with the `xyplot` from the `lattice` package as (essentially)

```
> xyplot(normexam ~ standLRT | sex * type, Exam, type = c("g", "p", "smooth"))
```

The formula would be read as “plot `normexam` by `standLRT` given `sex` by (school) `type`”. A few other arguments were added in the actual call to make the axis annotations more readable.

Figure 2 shows the even after accounting for a student's sex, pretest score and school type, there is considerable variation in the response. We may

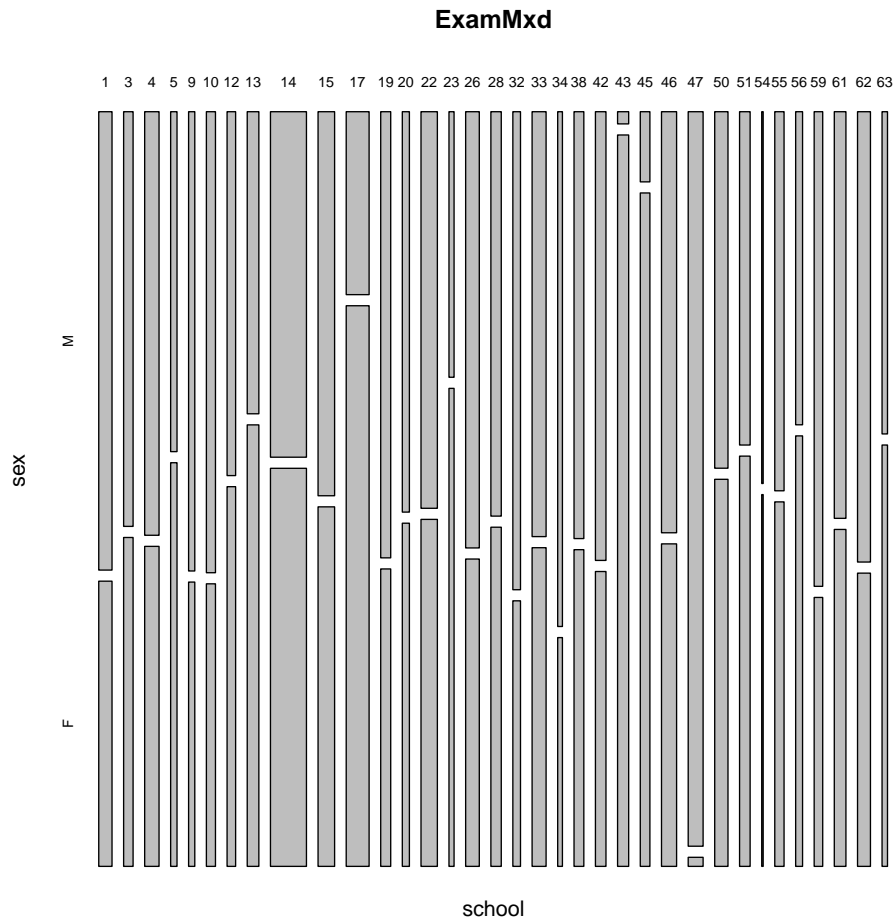


Figure 1: A mosaic plot of the sex distribution by school. The areas of the rectangles are proportional to the number of students of that sex from that school who took the exam. Schools with an unusually large or unusually small ratio of females to males are highlighted.

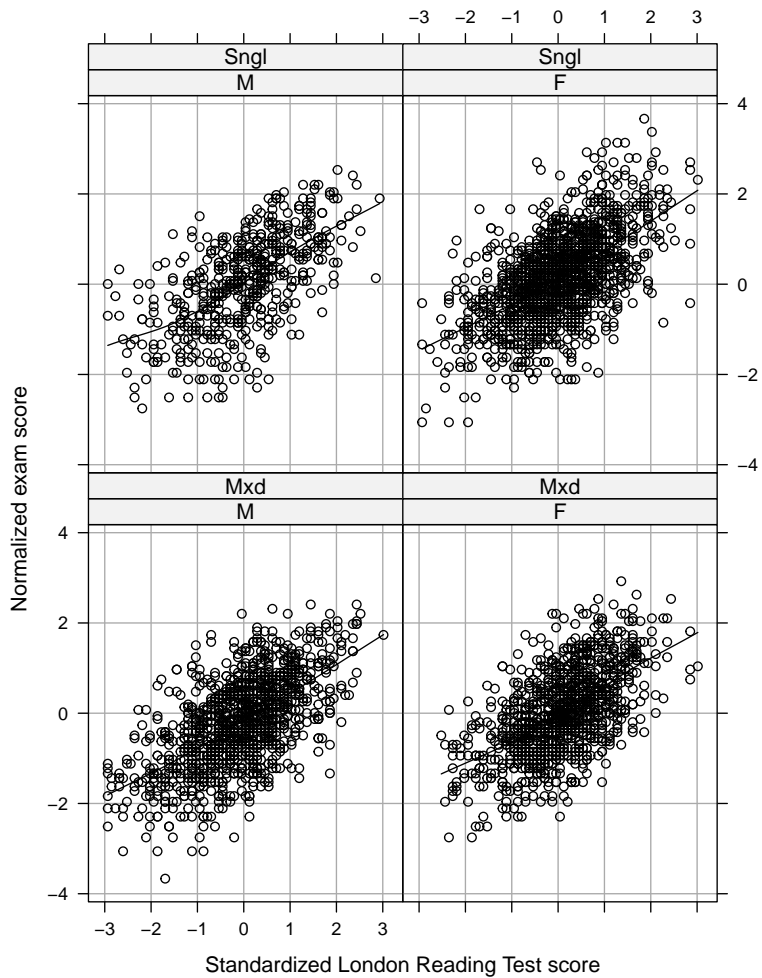


Figure 2: Normalized exam score versus pretest (Standardized London Reading Test) score for 4095 students from 65 schools in inner London. The panels on the left show the male students' scores; those on the right show the females' scores. The top row of panels shows the scores of students in single-sex schools and the bottom row shows the scores of students in mixed-sex schools. A scatterplot smoother line for each panel has been added to help visualize the trend.

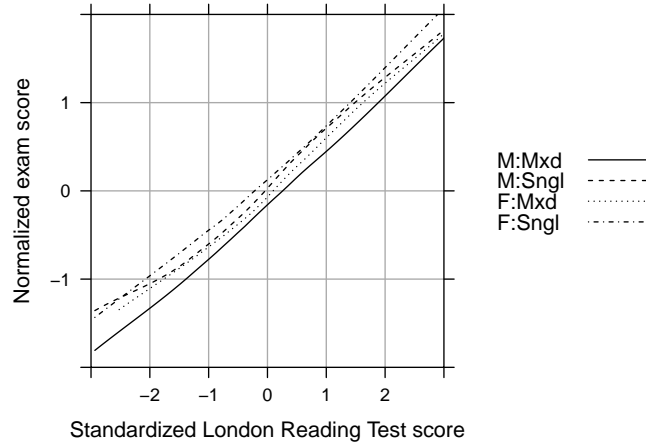


Figure 3: Overlaid scatterplot smoother lines of the normalized test scores versus the pretest (Standardized London Reading Test) score for female (F) and male (M) students in single-sex (Sngl) and mixed-sex (Mxd) schools.

attribute some of this variation to differences in schools but the fitted model indicates that most of the variation is unaccounted or “residual” variation.

In some ways the high level of residual variation obscures the pattern in the data. By removing the data points and overlaying the scatterplot smoothers we can concentrate on the relationships between the covariates. The call to `xyplot` is essentially

```
> xyplot(normexam ~ standLRT, Exam, groups = sex:type, type = c("g", "smooth"))
```

Figure 3 is a remarkable plot in that it shows nearly a perfect “main effects” relationship of the response with the three covariates and almost no interaction. It is rare to see real data follow a simple theoretical relationship so closely.

To elaborate, we can see that for each of the four groups the smoothed relationship between the exam score and the pretest score is close to a straight line and that the lines are close to being parallel. The only substantial deviation is in the smoothed relationship for the males in single-sex schools and this is the group with the fewest observations and hence the least precision in the estimated relationship. The lack of parallelism for this group is most apparent in the region of extremely low pretest scores where there are few observations and a single student who had a low pretest score and a moderate

post-test score can substantially influence the curve. Five or six such points can be seen in the upper left panel of Figure 2.

We should also notice the ordering of the lines and the spacing between the lines. The smoothed relationships for students in single-sex schools are consistently above those in the mixed-sex schools and, except for the region of low pretest scores described above, the relationship for the females in a given type of school is consistently above that for the males. Furthermore the distance between the female and male lines in the single-sex schools is approximately the same as the corresponding distance in the mixed-sex schools. We would summarize this by saying that there is a positive effect for females versus males and a positive effect for single-sex versus mixed-sex and no indication of interaction between these factors.

### 2.4.3 The effect of schools

We can check for patterns within and between schools by plotting the response versus the pretest by school. Because there appear to be differences in this relationship for single-sex versus mixed-sex schools and for females versus males we consider these separately.

In Figure 4 we plot the normalized exam scores versus the pretest score by school for female students in single-sex schools. The plot is produced as

```
> xyplot(normexam ~ standLRT | school, Exam,
+         type = c("g", "p", "r"),
+         subset = sex == "F" & type == "Sngl")
```

The "r" in the `type` argument adds a simple linear regression line to each panel.

The first thing we notice in Figure 4 is that school 48 is an anomaly because only two students in this school took the exam. Because within-school results based on only two students are unreliable, we will exclude this school from further plots (but we do include these data when fitting comprehensive models).

Although the regression lines on the panels can help us to look for variation in the schools, the ordering of the panels is, for our purposes, random. We recreate this plot in Figure 5 using

```
> xyplot(normexam ~ standLRT | school, Exam, type = c("g", "p", "r"),
+         subset = sex == "F" & type == "Sngl" & school != 48,
+         index.cond = function(x, y) coef(lm(y ~ x))[1])
```

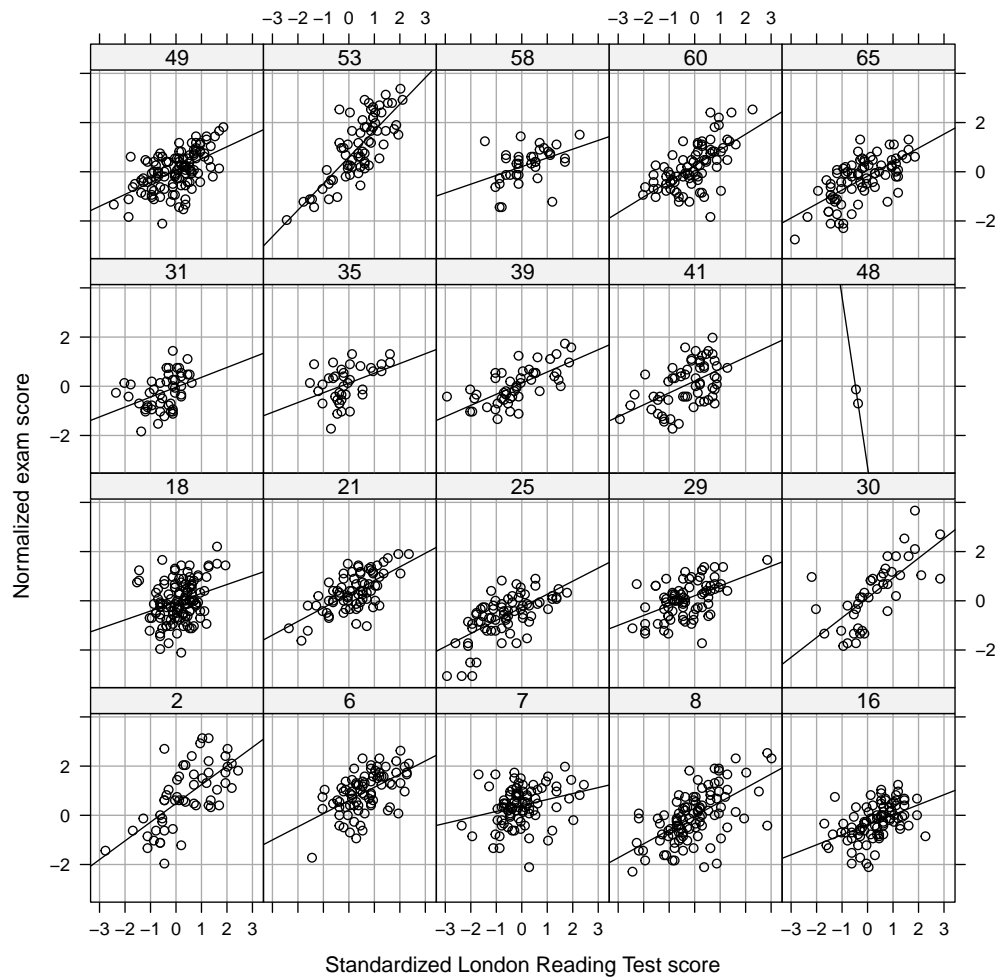


Figure 4: Normalized exam scores versus pretest (Standardized London Reading Test) score by school for female students in single-sex schools.

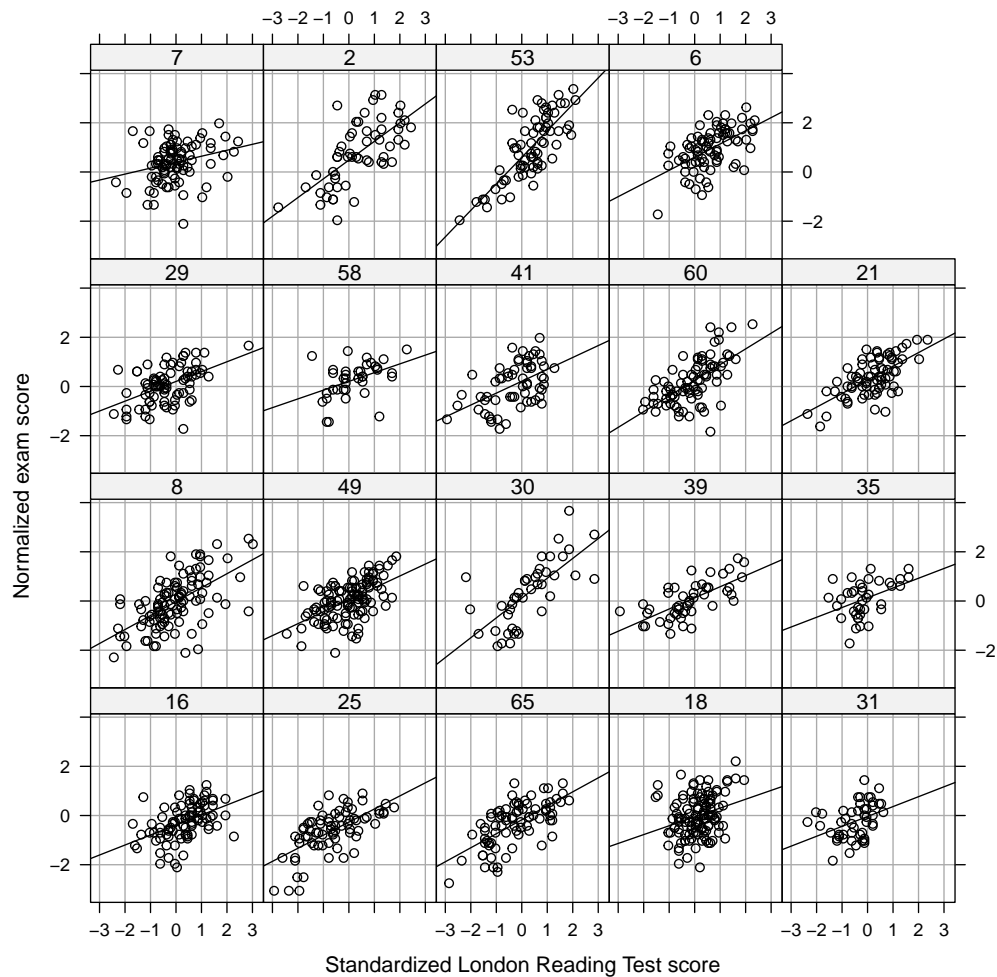


Figure 5: Normalized exam scores versus pretest (Standardized London Reading Test) score by school for female students in single-sex schools. School 48 where only two students took the exam has been eliminated and the panels have been ordered by increasing intercept (predicted normalized score for a pretest score of 0) of the regression line.

so that the panels are ordered (from left to right starting at the bottom row) by increasing intercept for the regression line (i.e. by increasing predicted exam score for a student with a pretest score of 0).

Alternatively, we could order the panels by increasing slope of the within-school regression lines, as in Figure 6.

Although it is informative to plot the within-school regression lines we need to assess the variability in the estimates of the coefficients before concluding if there is “significant” variability between schools. We can obtain the individual regression fits with the `lmList` function

```
> show(ExamFS <- lmList(normexam ~ standLRT | school, Exam,
+ subset = sex == "F" & type == "Sngl" & school != 48))
Call: lmList(formula = normexam ~ standLRT | school, data = Exam, subset = sex == "F" & type == "Sngl")
Coefficients:
  (Intercept) standLRT
 2  0.48227991 0.7612884
 6  0.60321439 0.5353444
 7  0.39852689 0.2422785
 8 -0.02519463 0.5674053
16 -0.38564292 0.4069399
18 -0.05733995 0.3593830
21  0.26872018 0.5544939
25 -0.26779146 0.5320575
29  0.20442314 0.4005158
30  0.11885028 0.8059021
31 -0.03922548 0.4022838
35  0.13173022 0.3966535
39  0.12754208 0.4525918
41  0.21249712 0.4834107
49  0.04747055 0.4845568
53  0.59370349 1.0769781
58  0.20707724 0.3557839
60  0.25196603 0.6378090
65 -0.17490019 0.5684592

Degrees of freedom: 1375 total; 1337 residual
Residual standard error: 0.7329521
```

and compare the confidence intervals on these coefficients.

```
> plot(confint(ExamFS, pool = TRUE), order = 1)

> show(ExamMS <- lmList(normexam ~ standLRT | school, Exam,
+ subset = sex == "M" & type == "Sngl"))
Call: lmList(formula = normexam ~ standLRT | school, data = Exam, subset = sex == "M" & type == "Sngl")
Coefficients:
  (Intercept) standLRT
11  0.26596312 0.4586355
24  0.17773174 0.3976156
27  0.03518861 0.5728684
36 -0.20691842 0.4383453
37 -0.48522245 0.2382739
```

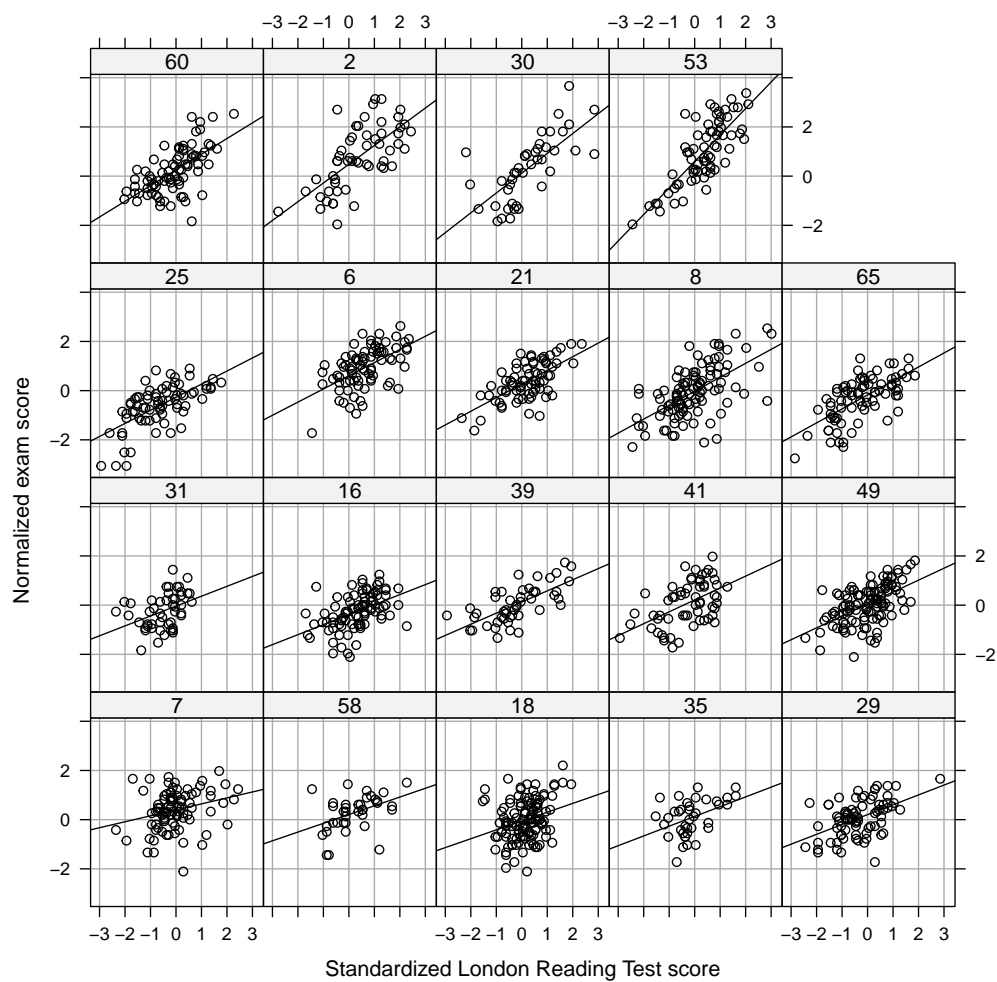


Figure 6: Normalized exam scores versus pretest (Standardized London Reading Test) score by school for female students in single-sex schools. School 48 has been eliminated and the panels have been ordered by increasing slope of the within-school regression lines.

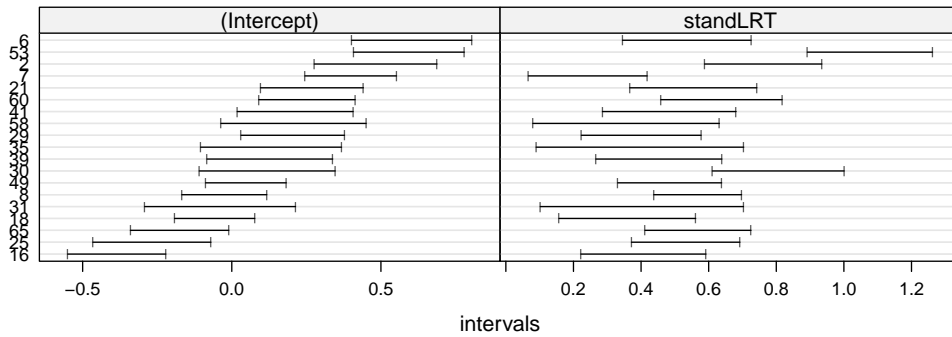


Figure 7: Confidence intervals on the coefficients of the within-school regression lines for female students in single-sex schools. School 48 has been eliminated and the schools have been ordered by increasing estimated intercept.

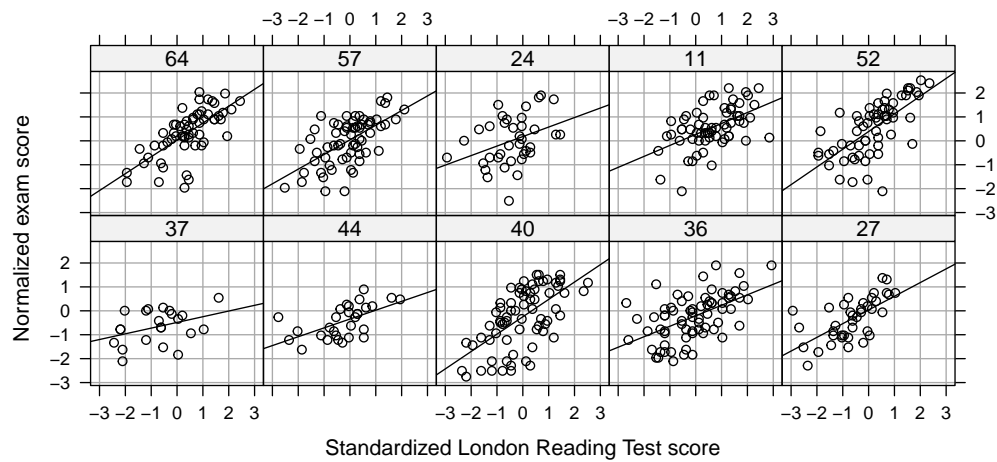


Figure 8: Normalized exam scores versus pretest (Standardized London Reading Test) score by school for male students in single-sex schools.

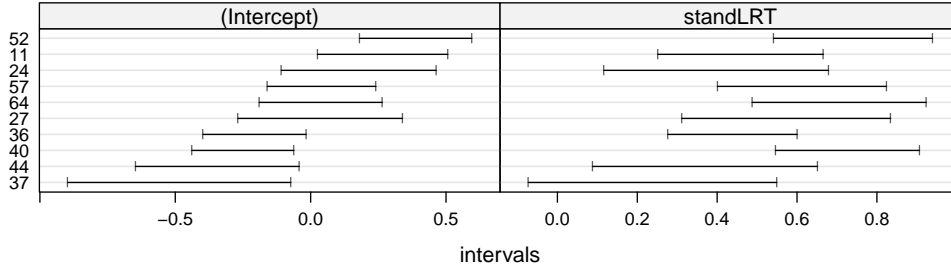


Figure 9: Confidence intervals on the coefficients of the within-school regression lines for female students in single-sex schools. School 48 has been eliminated and the schools have been ordered by increasing estimated intercept.

```
40 -0.25019842 0.7262845
44 -0.34440523 0.3696269
52  0.38803903 0.7402692
57  0.04055154 0.6123692
64  0.03749455 0.7055239
```

```
Degrees of freedom: 513 total; 493 residual
Residual standard error: 0.8082068
```

The corresponding plot of the confidence intervals is shown in Figure 9.

For the mixed-sex schools we can consider the effect of the pretest score and sex in the plot (Figure 10) and in the separate model fits for each school.

```
> show(ExamM <- lmList(normexam ~ standLRT + sex | school, Exam,
+                      subset = type == "Mxd" & !school %in% c(43,47,54)))
Call: lmList(formula = normexam ~ standLRT + sex | school, data = Exam, subset = type == "Mxd" & !sc
Coefficients:
  (Intercept) standLRT      sexF
1  0.24770238 0.7044798  0.355719842
3  0.58030950 0.5843480 -0.057223931
4 -0.16739321 0.7372405  0.402829332
5  0.36213174 0.6695127 -0.183238345
9 -0.32487665 0.3961812  0.222382659
10 -0.45139239 0.2972074  0.330917735
12 -0.26201220 0.4200265  0.378359201
13 -0.27196976 0.6037392  0.196013604
14 -0.28741229 0.5966633  0.202122649
15 -0.30963145 0.7370363  0.144392527
17 -0.30553035 0.4905235  0.156646395
19 -0.22542808 0.7214611  0.385544198
20  0.25120209 0.5187894  0.041342515
22 -0.50744197 0.5206371  0.089368918
23 -0.51727825 0.3873051 -0.189332381
```

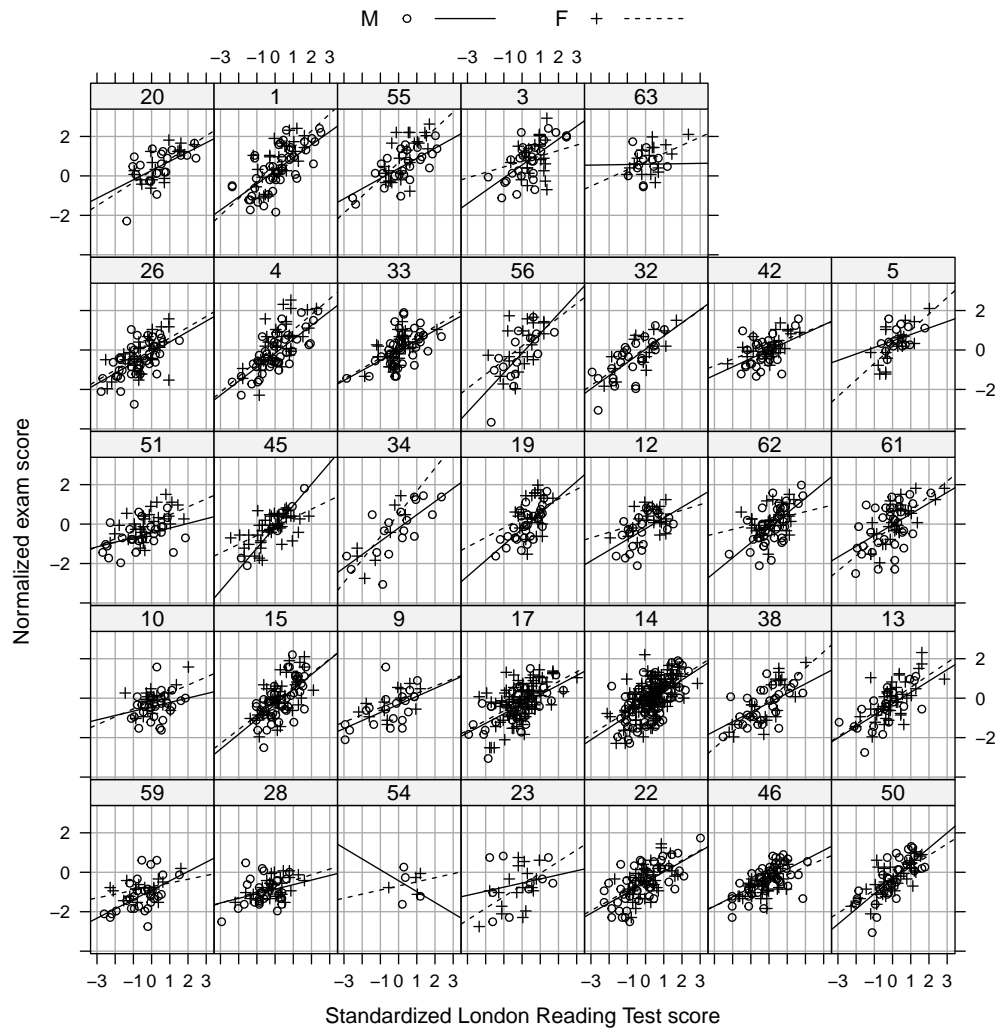


Figure 10: Normalized exam scores versus pretest score by school and sex for students in mixed-sex schools.

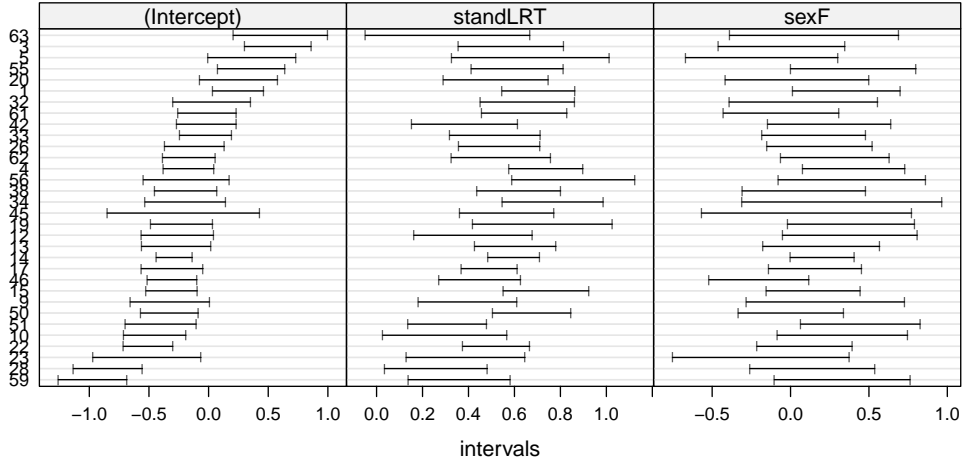


Figure 11: Confidence intervals on the coefficients of the within-school regression lines for female students in single-sex schools. School 48 has been eliminated and the schools have been ordered by increasing estimated intercept.

|    |             |           |              |
|----|-------------|-----------|--------------|
| 26 | -0.11870804 | 0.5342909 | 0.185283146  |
| 28 | -0.84451962 | 0.2583861 | 0.138672637  |
| 32 | 0.02596084  | 0.6560569 | 0.082029123  |
| 33 | -0.02539396 | 0.5148927 | 0.147967544  |
| 34 | -0.19582273 | 0.7662681 | 0.327656212  |
| 38 | -0.19255275 | 0.6184554 | 0.084872081  |
| 42 | -0.01913469 | 0.3827088 | 0.246533297  |
| 45 | -0.21212351 | 0.5665400 | 0.102317128  |
| 46 | -0.30554555 | 0.4491187 | -0.201294993 |
| 50 | -0.32718434 | 0.6752947 | 0.001906973  |
| 51 | -0.40150931 | 0.3076539 | 0.445100548  |
| 55 | 0.35743002  | 0.6118447 | 0.400034477  |
| 56 | -0.18744293 | 0.8558783 | 0.391178135  |
| 59 | -0.97233088 | 0.3594417 | 0.329480168  |
| 61 | -0.01215078 | 0.6428683 | -0.060024560 |
| 62 | -0.16445110 | 0.5411566 | 0.283642476  |
| 63 | 0.60216184  | 0.3091657 | 0.150390337  |

Degrees of freedom: 2018 total; 1922 residual  
Residual standard error: 0.7273955

The confidence intervals for these separately fitted models, shown in Figure 11 indicate differences in the intercepts and possibly differences in the slopes with respect to the pretest scores. However, there is not a strong indication of variation by school in the effect of sex.

## 2.5 Multilevel models for the exam data

We begin with a model that has a random effects for the intercept by school plus additive fixed effects for the pretest score, the student's sex and the school type.

```
> (Em3 <- lmer(normexam ~ standLRT + sex + type + (1|school), Exam))
Linear mixed model fit by REML ['lmerMod']
Formula: normexam ~ standLRT + sex + type + (1 | school)
Data: Exam
REML criterion at convergence: 9345.384
Random effects:
  Groups   Name                Std.Dev.
  school   (Intercept) 0.2905
  Residual                    0.7500
Number of obs: 4059, groups: school, 65
Fixed Effects:
(Intercept)      standLRT          sexF      typeSngl
      -0.1677          0.5598          0.1660          0.1655
```

Our data exploration indicated that the slope with respect to the pretest score may vary by school. We can fit a model with random effects by school for both the slope and the intercept as

```
> (Em4 <- lmer(normexam ~ standLRT + sex + type + (standLRT|school), Exam))
Linear mixed model fit by REML ['lmerMod']
Formula: normexam ~ standLRT + sex + type + (standLRT | school)
Data: Exam
REML criterion at convergence: 9300.573
Random effects:
  Groups   Name                Std.Dev. Corr
  school   (Intercept) 0.2872
           standLRT   0.1228  0.58
  Residual                    0.7418
Number of obs: 4059, groups: school, 65
Fixed Effects:
(Intercept)      standLRT          sexF      typeSngl
      -0.1887          0.5541          0.1680          0.1764
```

and compare this fit to the previous fit with

```
> anova(Em3, Em4)
Data: Exam
Models:
Em3: normexam ~ standLRT + sex + type + (1 | school)
Em4: normexam ~ standLRT + sex + type + (standLRT | school)
      npar    AIC    BIC logLik -2*log(L)  Chisq Df Pr(>Chisq)
Em3     6 9337.5 9375.3 -4662.7   9325.5
Em4     8 9297.1 9347.6 -4640.6   9281.1 44.327  2 2.368e-10
```

There is a strong evidence of a significant random effect for the slope by school, whether judged by AIC, BIC or the p-value for the likelihood ratio test.

The p-value for the likelihood ratio test is based on a  $\chi^2$  distribution with degrees of freedom calculated as the difference in the number of parameters in the two models. Because one of the parameters eliminated from the full model in the submodel is at its boundary the usual asymptotics for the likelihood ratio test do not apply. However, it can be shown that the p-value quoted for the test is conservative in the sense that it is an upper bound on the p-value that would be calculated say from a parametric bootstrap.

Having an upper bound of  $1.9 \times 10^{-10}$  on the p-value can be regarded as “highly significant” evidence of the utility of the random effect for the slope by school.

We could also add a random effect for the student’s sex by school

```
> (Em5 <- lmer(normexam ~ standLRT + sex + type + (standLRT + sex|school), Exam))
Linear mixed model fit by REML ['lmerMod']
Formula: normexam ~ standLRT + sex + type + (standLRT + sex | school)
Data: Exam
REML criterion at convergence: 9299.956
Random effects:
  Groups   Name                Std.Dev. Corr
  school   (Intercept)  0.27550
           standLRT    0.12280  0.62
           sexF        0.02913  0.70 -0.14
Residual   0.74177
Number of obs: 4059, groups: school, 65
Fixed Effects:
(Intercept)    standLRT         sexF    typeSngl
   -0.1895         0.5541         0.1698         0.1762
optimizer (nloptwrap) convergence code: 0 (OK) ; 0 optimizer warnings; 1 lme4 warnings
```

Notice that the estimate of the variance of the `sexM` term is essentially zero so there is no need to test the significance of this variance component. We proceed with the analysis of `Em4`.

### 3 Growth curve model for repeated measures data

```
> str(Oxboys)
'data.frame':      234 obs. of  4 variables:
 $ Subject : Factor w/ 26 levels "1","10","11",...: 1 1 1 1 1 1 1 1 1 12 ...
 $ age     : num  -1 -0.7479 -0.463 -0.1643 -0.0027 ...
 $ height  : num  140 143 145 147 148 ...
 $ Occasion: Factor w/ 9 levels "1","2","3","4",...: 1 2 3 4 5 6 7 8 9 1 ...
- attr(*, "ginfo")=List of 7
 ..$ formula      :Class 'formula' language height ~ age | Subject
 .. .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
 ..$ order.groups: logi TRUE
 ..$ FUN          :function (x)
```

```

.. ..- attr(*, "source")= chr "function (x) max(x, na.rm = TRUE)"
..$ outer      : NULL
..$ inner      : NULL
..$ labels     :List of 2
.. ..$ age     : chr "Centered age"
.. ..$ height  : chr "Height"
..$ units      :List of 1
.. ..$ height  : chr "(cm)"

> system.time(mX1 <- lmer(height ~ age + I(age^2) + I(age^3) + I(age^4) + (age + I(age^2)|Subject),
+              Oxboys))

  user system elapsed
0.041  0.000  0.041

> summary(mX1)
Linear mixed model fit by REML ['lmerMod']
Formula: height ~ age + I(age^2) + I(age^3) + I(age^4) + (age + I(age^2) |
  Subject)
Data: Oxboys

REML criterion at convergence: 627.9

Scaled residuals:
    Min       1Q   Median       3Q      Max
-2.71332 -0.54610  0.00294  0.54290  2.65872

Random effects:
 Groups Name      Variance Std.Dev. Corr
 Subject (Intercept) 64.0361  8.0023
          age         2.8641  1.6924  0.61
          I(age^2)    0.6743  0.8212  0.22 0.66
Residual                0.2174  0.4662
Number of obs: 234, groups: Subject, 26

Fixed effects:
              Estimate Std. Error t value
(Intercept) 149.0189    1.5704  94.894
age          6.1742     0.3565  17.319
I(age^2)     1.1282     0.3514   3.210
I(age^3)     0.4539     0.1625   2.794
I(age^4)    -0.3769     0.3002  -1.256

Correlation of Fixed Effects:
          (Intr) age    I(g^2) I(g^3)
age       0.572
I(age^2)  0.076  0.264
I(age^3) -0.001 -0.340  0.025
I(age^4)  0.021  0.016 -0.857 -0.021

> system.time(mX2 <- lmer(height ~ poly(age, 4) + (age + I(age^2)|Subject), Oxboys))

  user system elapsed
0.041  0.000  0.041

> summary(mX2)

```

```
Linear mixed model fit by REML ['lmerMod']
Formula: height ~ poly(age, 4) + (age + I(age^2) | Subject)
Data: Oxboys
```

```
REML criterion at convergence: 616.9
```

```
Scaled residuals:
    Min      1Q   Median      3Q      Max
-2.71331 -0.54609  0.00294  0.54290  2.65871
```

```
Random effects:
Groups Name Variance Std.Dev. Corr
Subject (Intercept) 64.0357  8.0022
        age         2.8641  1.6924  0.61
        I(age^2)    0.6743  0.8211  0.22 0.66
Residual          0.2174  0.4662
Number of obs: 234, groups: Subject, 26
```

```
Fixed effects:
              Estimate Std. Error t value
(Intercept) 149.5198    1.5903  94.019
poly(age, 4)1  64.5409    3.3278  19.394
poly(age, 4)2   4.2032    1.0236   4.106
poly(age, 4)3   1.2908    0.4663   2.768
poly(age, 4)4  -0.5855    0.4663  -1.256
```

```
Correlation of Fixed Effects:
              (Intr) p(,4)1 p(,4)2 p(,4)3
poly(ag,4)1  0.631
poly(ag,4)2  0.230  0.583
poly(ag,4)3  0.000  0.000  0.000
poly(ag,4)4  0.000  0.000  0.000  0.000
```

## 4 Cross-classification model

```
> str(ScotsSec)
'data.frame':      3435 obs. of  6 variables:
 $ verbal : num  11 0 -14 -6 -30 -17 -17 -11 -9 -19 ...
 $ attain : num  10 3 2 3 2 2 4 6 4 2 ...
 $ primary: Factor w/ 148 levels "1","2","3","4",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ sex    : Factor w/ 2 levels "M","F": 1 2 1 1 2 2 2 1 1 1 ...
 $ social : num  0 0 0 20 0 0 0 0 0 0 ...
 $ second : Factor w/ 19 levels "1","2","3","4",...: 9 9 9 9 9 9 1 1 9 9 ...

> system.time(mS1 <- lmer(attain ~ sex + (1|primary) + (1|second), ScotsSec))
   user system elapsed
 0.040  0.000  0.041

> summary(mS1)
Linear mixed model fit by REML ['lmerMod']
Formula: attain ~ sex + (1 | primary) + (1 | second)
Data: ScotsSec

REML criterion at convergence: 17127.9
```

```

Scaled residuals:
  Min       1Q   Median       3Q      Max
-2.2376 -0.8469 -0.1219  0.9035  2.2001

Random effects:
 Groups   Name      Variance Std.Dev.
primary  (Intercept)  1.1096   1.053
second   (Intercept)  0.3697   0.608
Residual                   8.0551   2.838
Number of obs: 3435, groups: primary, 148; second, 19

Fixed effects:
              Estimate Std. Error t value
(Intercept)  5.25515    0.18432  28.511
sexF         0.49851    0.09825   5.074

Correlation of Fixed Effects:
      (Intr)
sexF  -0.264

```

## 5 Session Info

```
> toLatex(sessionInfo())
```

- R version 4.6.0 (2026-04-24), x86\_64-pc-linux-gnu
- Locale: LC\_CTYPE=en\_US.UTF-8, LC\_NUMERIC=C, LC\_TIME=en\_US.UTF-8, LC\_COLLATE=en\_US.UTF-8, LC\_MONETARY=en\_US.UTF-8, LC\_MESSAGES=en\_US.UTF-8, LC\_PAPER=en\_US.UTF-8, LC\_NAME=C, LC\_ADDRESS=C, LC\_TELEPHONE=C, LC\_MEASUREMENT=en\_US.UTF-8, LC\_IDENTIFICATION=C
- Time zone: Etc/UTC
- TZcode source: system (glibc)
- Running under: Ubuntu 24.04.4 LTS
- Matrix products: default
- BLAS: /usr/lib/x86\_64-linux-gnu/openblas-pthread/libblas.so.3
- LAPACK: /usr/lib/x86\_64-linux-gnu/openblas-pthread/libopenblas-p0.3.26.so ; LAPACK version3.12.0

- Base packages: base, datasets, graphics, grDevices, methods, stats, utils
- Other packages: lattice 0.22-9, lme4 2.0-1, Matrix 1.7-5, mlmRev 1.0-9
- Loaded via a namespace (and not attached): boot 1.3-32, buildtools 1.0.0, cli 3.6.6, compiler 4.6.0, evaluate 1.0.5, grid 4.6.0, knitr 1.51, maketools 1.3.2, MASS 7.3-65, minqa 1.2.8, nlme 3.1-169, nloptr 2.2.1, rbibutils 2.4.1, Rcpp 1.1.1-1.1, Rdpack 2.6.6, reformulas 0.4.4, rlang 1.2.0, splines 4.6.0, sys 3.4.3, tools 4.6.0, xfun 0.58